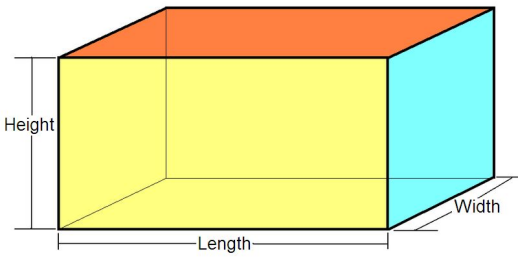
S201 – Fall 2020-2021
Take-Home Exam 2
Due November 15th, Sunday, 23:55 (Sharp Deadline)

Description

The aim of this homework is to practice on parametric functions and if statements. The use of if statements is due to the nature of the problem (I mean you cannot finish this homework without using if statements). However, the use of functions is somehow enforced. That means it is possible to accomplish this homework without using functions, but it is a must to use them. The details about the use of functions in this homework are given later in this document.

In this homework, you will write a C++ program that calculates and displays the area and the volume information of two cuboids, which are identified by their colors. Then, the program will test whether two cuboids are cubes and finally compare these cuboids to each other in terms of area and volume.



A cuboid (see the figure on the right) is a solid geometric figure bounded by six rectangular faces (a rectangular box). All angles of a cuboid are right angles and opposite faces are equal. For more information, you may refer to Wikipedia at http://en.wikipedia.org/wiki/Cuboid.

The program starts by displaying a welcome message. Then a prompt is displayed to direct the user of the program to enter two colors for the cuboids. These colors must be different and will be used to refer to the cuboids later in the program (e.g. "red cuboid is ...", "yellow cuboid is ..."). After the colors are entered, your program should ask the user to input length, width, and height values (dimensions) for the cuboids by first displaying appropriate prompts using the colors of the cuboids. After these values are entered, for each cuboid your program should check whether the length, width, and height values for that cuboid are entered correctly or not (see Inputs and Input Check section for details).

Program Flow

For each cuboid, your program:

- should display a message indicating whether the cuboid is a cube or not. If all three dimensions (length, width, and height) of a cuboid are the same, then it is called a cube.
- should calculate the area and volume of the cuboid and display the results on the screen.

After the above steps are done and there is no problem in any input format, the program will compare the two cuboids in terms of area and volume.

Note: If the dimensions of any cuboid are not entered correctly, your program should display an error message and should not perform any other operation for any of the cuboids. See the Sample Runs section to understand the flow of the program much better.

Your take-home exams will be automatically graded using GradeChecker, so it is **very important to satisfy the exact same output given in the sample runs.** You can utilize GradeChecker (http://sky.sabanciuniv.edu:8080/GradeChecker/) to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the take-home exam (**all** files, including *strutils*). Additionally, you should submit all of your files to SUCourse (**all** files, including *strutils*) **without zipping** them. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_THEnumber.cpp" (all lowercase letters). Please check the submission procedures of the take-home exam, which are listed at the end of this document.

Inputs and Input Check

There are some inputs to your program. These are (i) color, (ii) length, (iii) width, and (iv) height of the cuboids. You should have two sets of such variables and inputs, one set for each cuboid. The color is of type string. The length, width, and height are of double type (real numbers). The order of inputs and outputs are explained above. See the Sample Runs section for some examples of double type (real numbers). The order of inputs and outputs are explained above. See the Sample Runs section for some examples.

In this homework you **have to** perform some input checks as detailed below:

- Any string entered as a color will be accepted, although the entered string may not be meaningful or not a color name. However, the colors of two cuboids cannot be the same. If they are the same, then the program should come to the end without asking for the dimensions of the cuboids and making other operations. The equality check for the colors will be performed in case insensitive manner such that uppercase and lowercase letters will be considered as same letters. For example, red and REd will be considered as same colors (In order to do this, you need to use some string utility functions). You may assume that each color contains only one word.
- Length, width and height values must be greater than zero. For cuboids, if one or more of these inputs is/are less than or equal to zero, then your program should display an error message. In this message, you have to specify which of the cuboids has an input problem by using its color.
- In the error message, you do not need to differentiate which dimension is entered wrong. A generic error message (for example, “invalid entry for *red* cuboid”) is sufficient, please see sample runs.

You are not expected to (and please do not try to) re-input a variable if an erroneous case described above occurs. Such functionality requires loops that we have not seen yet. Please see Sample Runs below for some examples of wrong inputs.

All of the input operations (**`cin`** statements) will be in the main function. However, you are going to pass the inputs to several functions. Input check for the colors will be performed in the main function. On the other hand, input check operation for the dimensions will be performed in a user-defined function. See Use of Functions section for more information.

Computations and the output

Your program should calculate the area and volume of cuboids using following formulas:

*Volume = height * width * length*
*Area = 2 * (height * width + width * length + height * length)*

As the output, for each cuboid, first you have to display whether the cuboid is a cube or not. After that, you have to display results of area and volume. Example output can be seen in the Sample Runs section below.

Since you are dealing with real numbers in this program, the output may not be so tidy for some values. Default real number output may be in scientific notation or several digits may be displayed after the decimal point. In order to have tidy real number output, there is a mechanism to fix the amount of digits to be displayed after the decimal point. To do so, first you have to include the **`iomanip`** header file at the beginning of you program (preferably after **`iostream`**) by writing

```
#include <iomanip>
```

Then, insert those two lines as the first statements of the main function:

```
cout.setf(ios::fixed);
cout.precision(2);
```

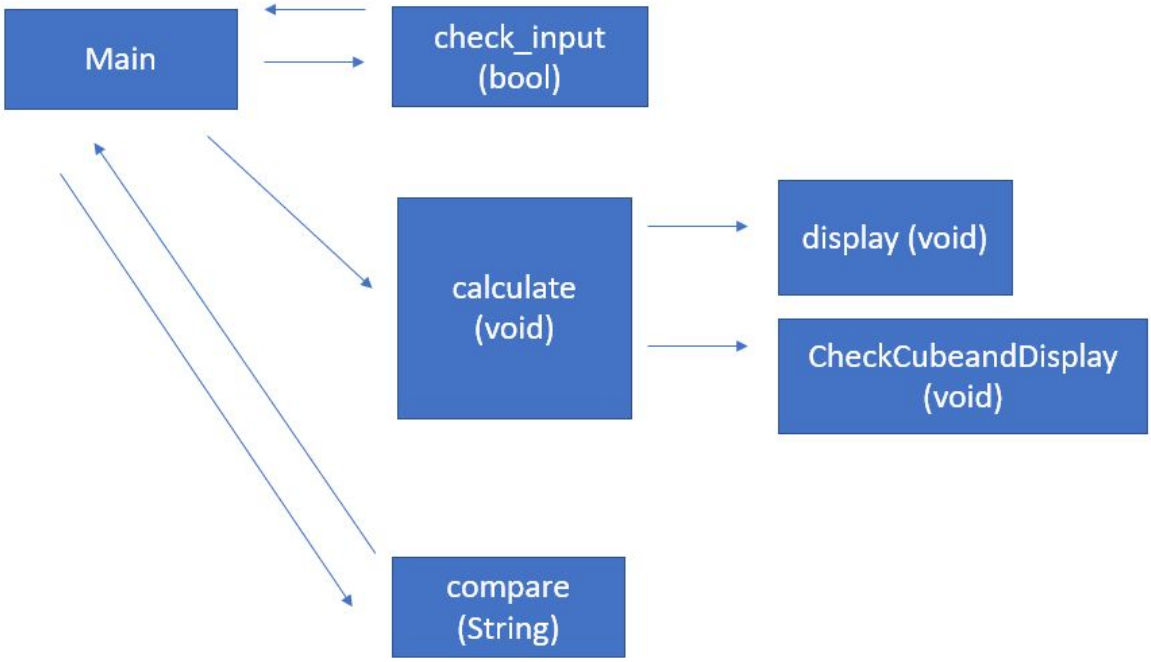
This piece of code will cause the whole program to display floating point numbers (real numbers) in a more readable format (with two fractional digits). It is sufficient to have this code only in the main function; you do not need to repeat it in other functions.

Use of Functions (EXTREMELY IMPORTANT)

In the first homework you were not supposed to implement any functions. However, in this homework you are expected to (actually you have to) use functions. The guidance about using functions in this homework is below.

You have to follow the specifications below for function declarations and callings. The grading criteria will include proper use of these parametric functions. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use. Unnecessary code duplication will cause grade reduction as well.

A total of five user-defined functions (other than main) must be implemented.



Function1 (Boolean) check_input: This is a parametric boolean function that performs the check for the validity of the dimensions of a cuboid. This function is going to be a parametric one, too. These parameters will be the color, width, height and the length information of the cuboid. As mentioned above, the length, width and height values of a specific cuboid must be positive real numbers. If not, this function should display an error message for this particular cuboid by specifying its color, and should return false to the main function otherwise should return true. The main function should decide whether to continue the program or not according to this boolean value. This function must be called once for each cuboid in the main function.

Function2 (void) *checkCubeandDisplay*: You will write a function that checks whether a specific cuboid (identified by its color) is a cube or not, and displays this information on the screen. This is going to be a parametric function. The parameters should be used to pass the color, width, height, and length information of the cuboid to this function. This function is to be called in Function 4.

Function3 (void) *display*: You will write another function to display the result of the area and volume of a cuboid. This function will display the area and volume of a cuboid together with its color. **This function is NOT going to calculate area and volume**; it is just going to display them. Thus, do not have any calculations in this function. The data to be output will be passed into this function as parameters. As you may see from the sample runs, the lines that display outputs of calculations are almost the same

except color, area, and volume of the cuboids (this sentence was a hint about the parameters of this function). This function is to be called in the function that you make the calculations (Function4).

Function4 (void) *calculate*: You are going to write another function that will perform area and volume calculations for a single cuboid and then display them. This function is going to be a parametric one, too. These parameters will be the color, width, height, length, **area and volume** information of the cuboid (6 parameters). The most critical point here is that the area and volume parameters need to be passed as reference parameters. In other words, the area and volume variables must be defined in the main function first for each cuboid and passed to this function (Function4) with the reference parameter. Then, after the necessary calculations are made, the results obtained will be assigned to the variables corresponding volume and area. Using the parameters of the function, you will perform calculations according to the given formulas. In order to display the area and volume information together with the color of the cuboid, you have to call Function2 and Function3 with proper arguments (do not have any cout statements in Function 4). Notice that, Function2 and Function3 should be called in Function4 and Function4 must be called in main function.

Function5 (string) *compare*: This function will compare the area and volume of cuboids and return the required output as a string to the main function. This function will be called from the main function and parameters should be the colors of cuboids, the desired metric (area or volume) as string and the value of this metric for both cuboids, 5 parameters in total. This function should be called twice through main and will compare the volumes of cuboids if the metric is selected as "Volume", and if it is selected as "Area" it will compare the areas of cuboids.

This function will return 3 different outputs according to comparisons. For example, we have two cuboids, green and red. If metric parameter is selected as “Area”, in this case, 3 different outputs are as follows:

- Red has the largest area.
- Green has the largest area.
- Two cuboids have the same area.

Remind that, if the metric was chosen as the volume instead of the area, the message would also write the volume accordingly.

Needless to say, you have to name these five functions using meaningful identifiers, not as Function1, Function2, etc.

Other than these five functions, you will also have the **`main`** function. The welcome message display, prompts for input, and input operations (**`cin`** statements) will be in **`main`**. You will also check the inequality of two color inputs also in **`main`**.

No abrupt program termination please!

Especially during the input checks, you may want to stop the execution of the program at a specific place in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

IMPORTANT!

If your code does not compile, then you will get **zero**. Please be careful about this and double check your code before submission.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the mentioned format.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some points in the best scenario.

Sample Input & Output

Below, we provide some sample runs of the program that you will develop. The italic and bold phrases are inputs taken from the user.

Sample Run 1

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: RED RED
Color names cannot be the same, good bye...
```

Sample Run 2

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: red ReD
Color names cannot be the same, good bye...
```

Sample Run 3

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: GREEN RED
Please enter length, width and height of the GREEN cuboid: 0 23 12
Invalid entry for GREEN cuboid. Dimensions should be positive real numbers!
```

Sample Run 4

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: red green
Please enter length, width and height of the red cuboid: 24 A 24
Invalid entry for red cuboid. Dimensions should be positive real numbers!
```

Sample Run 5

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: RED GREEN
Please enter length, width and height of the RED cuboid: 1 1 1
Please enter length, width and height of the GREEN cuboid: 2 -2 3
Invalid entry for GREEN cuboid. Dimensions should be positive real numbers!
```

Sample Run 6

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: RED GREEN
Please enter length, width and height of the RED cuboid: 1 1 1
Please enter length, width and height of the GREEN cuboid: 2 2 2
Results for the RED cuboid:
Area: 6.00
Volume: 1.00
RED cuboid is a cube.

Results for the GREEN cuboid:
Area: 24.00
Volume: 8.00
GREEN cuboid is a cube.

Comparison of the two cuboids:
GREEN has the largest volume.
GREEN has the largest area.
```

Sample Run 7

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: RED GREEN
Please enter length, width and height of the RED cuboid: 2 2 2
Please enter length, width and height of the GREEN cuboid: 1 1 1
Results for the RED cuboid:
Area: 24.00
Volume: 8.00
RED cuboid is a cube.

Results for the GREEN cuboid:
Area: 6.00
Volume: 1.00
GREEN cuboid is a cube.

Comparison of the two cuboids:
RED has the largest volume.
RED has the largest area.
```

Sample Run 8

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: red Blue
Please enter length, width and height of the red cuboid: 1.2 1.2 1.2
Please enter length, width and height of the Blue cuboid: 1.2 1.2 1.2
Results for the red cuboid:
Area: 8.64
Volume: 1.73
red cuboid is a cube.

Results for the Blue cuboid:
Area: 8.64
Volume: 1.73
Blue cuboid is a cube.

Comparison of the two cuboids:
Two cuboids have the same volume.
Two cuboids have the same area.
```

Sample Run 9

```
Hello! This program compares two cuboids...
Please enter colors for the two cuboids: yellow black
Please enter length, width and height of the yellow cuboid: 20 20 1
Please enter length, width and height of the black cuboid: 10 10 10
Results for the yellow cuboid:
Area: 880.00
Volume: 400.00
yellow cuboid is not a cube.

Results for the black cuboid:
Area: 600.00
Volume: 1000.00
black cuboid is a cube.

Comparison of the two cuboids:
black has the largest volume.
yellow has the largest area.
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all take-home exams, unless otherwise noted.

- **How to get help?**

You can use GradeChecker (http://sky.sabanciuniv.edu:8080/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the course website.

- **What and Where to Submit**

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using Xcode (macOS users).

It'd be a good idea to write your name and last name in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:

// Inanc Arin

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:
    - Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
    - Name your cpp file that contains your program as follows:
      "SUCourseUsername_THEnumber.cpp"
    - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**). For example, if your SUCourse username is "**altop**", then the file name should be: **altop_the1.*cpp*** (please only use lowercase letters).
    - Do not add any other character or phrase to the file name.
- Please make sure that this file is the latest version of your take-home exam program.
- Submit your work **through SUCourse only**! You can use GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).
- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

Grading, Review and Objections

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your take-home exam, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.
- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the Syllabus or on the course website.

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your take-home exam from the email address provided in the comment section of your announced take-home exam grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the take-home exam tab to see the problem with your take-home exam.
- Download the file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Anıl Özdemir & Barış Altop & Gülşen Demiröz